

Министерство науки и высшего образования РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Курский государственный университет»

Кафедра программного  
обеспечения и администрирования  
информационных систем

Направление подготовки  
математическое обеспечение и  
администрирование  
информационных систем

Форма обучения очная

**Отчет**  
**по лабораторной работе №4**  
**«Обработка массивов. Числа**  
**Фибоначчи»**

Выполнил:

студент группы 213

Водолад Д.В.

Проверил:

доцент кафедры ПОиАИС

Кривонос А.В.

Курск, 2021

### ***Цели работы:***

- 1) изучение принципов функционирования памяти и микропроцессора компьютера при выполнении операций над массивами данных;
- 2) приобретение навыков использования команд ассемблера, связанных с обработкой массивов;
- 3) получение представления об особенностях обработки данных, команд и режимах доступа к данным при обработке массивов.

### ***Задание:***

- 1) Написать программу на языке ассемблера, которая использует массивы и позволяет вычислить числа Фибоначчи в заданном диапазоне.
  - 1.1) В сегменте данных определить массив из 18 двухбайтовых ячеек с произвольным первоначальным значением, две двухбайтовые ячейки с нулевым первоначальным значением для размещения минимального и максимального элементов массива, соответственно.
  - 1.2) Вычислить первые 18 чисел Фибоначчи и поместить их в массив, обращаясь к нему как к одномерному массиву.
  - 1.3) Рассматривая имеющийся массив как двухмерный размера 3×6 (3 строки, 6 столбцов), найти наименьших из нечётных элементов второй строки и наибольший из чётных элементов четвёртого столбца.
  - 1.4) Полученные результаты поместить в соответствующие ячейки памяти.

### ***Текст программы***

```
d_s segment
    arr DW 18 dup(?)      ; массив
    i DW ?
    j DW ?
```

```
min DW 0
max DW 0
d_s ends
```

```
c_s segment
    assume ss:s_s, ds_d:ds_d, cs:c_s
begin:
```

```
    mov ax, d_s
    mov ds, ax
    xor ax, ax
```

```
    mov si, 2
    mov ax, si
    mov cx, 16
```

```
    mov arr, 0
    mov arr + 2, 1
```

```
FIB:          ; метка цикла
```

```
    mov si, ax
    shl si, 1; * 2
    sub si, 4
    mov bx, arr-si
    add si, 2
    add bx, arr-si
    add si, 2
    mov arr+si, bx
    inc ax
    mov min, bx
```

```
loop FIB      ; при помощи loop происходит увеличение счетчика в cx
```

```
    mov cx, 6
    mov i, 1
    mov j, cx
    mov ax, 1500
```

```
WH1:          ; метка цикла
```

```
    mov si, cx
    dec si
    shl si, 1; *2
    mov bx, arr + 12 + si
    mov dx, bx
    shr dx, 1
    jc IF1
```

```
loop WH1
jmp endWH1
```

```
IF1:          ; метка условия
cmp bx, ax
jl A1
loop WH1
jmp endWH1
```

```
A1:          ; метка условия
mov ax, bx
loop WH1
```

```
endWH1:      ; метка выхода из цикла
```

```
mov min, ax
xor bx, bx
xor dx, dx
; arr + stroka * razmer * i + j * razmer
; arr + 6 * 2 * i + 8 <=> 12 * si + 8
```

```
mov cx, 3
mov j, 1
mov i, cx
mov dx, 0
WH2:          ; метка цикла
mov si, cx
dec si
```

```
mov max, dx
mov ax, si
mov bx, 12
imul bx
mov dx, max
```

```
mov si, ax
```

```
mov bx, arr + 6 + si
mov ax, bx
shr ax, 1
jnc IF2
loop WH2
jmp endWH2
```

```
IF2:          ; метка условия
    cmp bx, dx
    jg A2
    loop WH2
    jmp endWH2
```

```
A2:          ; метка условия
    mov dx, bx
    loop WH2
```

```
endWH2:      ; метка выхода из цикла
```

```
    mov max, dx
```

```
    mov ah, 4ch
    int 21h
```

```
c_s ends
end begin
```

### ***Вывод:***

- 1) Я изучил принципы функционирования памяти и микропроцессора компьютера при выполнении операций над массивами данных;
- 2) приобрел навыки использования команд ассемблера, связанных с обработкой массивов;
- 3) получил представление об особенностях обработки данных, команд и режимах доступа к данным при обработке массивов.